

# YETI 2009 Tutorials: Day 1

## Underlying Event

### 1 Starting using Sherpa: Output one event

Typically running Sherpa is a two step process. First, Sherpa is run to initialise the process. During this initial run, Sherpa identifies and creates the relevant libraries needed for event generation. The user must then compile the libraries, by simply typing `./makelibs`. Then the user runs Sherpa again to perform the process integration and generate the events. For the rest of tutorials today and tomorrow, the necessary libraries have already been created, compiled and linked, and the integration results provided, as these steps can be time consuming for some processes. However, for this first run you must create your own libraries, and then we will look at the structure of a single event.

Copy the prepared Run.dat into a working directory,

```
mkdir intro
cp /mt/data-grid/yeti/setups/intro/Run.dat intro/
cd intro
```

and take a look at it. The **OUTPUT** flag has been set to **3** (Events+Info). This setting means that details about each event are displayed when Sherpa is run. Therefore, this is only suitable when generating one or two events. The default value for this flag is **2** (Info), and you will be using this later when you generate more events. The number of events to be generated has been set to **1**.

Back at the command line, run Sherpa using the command:

```
Sherpa
```

Then compile the libraries using:

```
./makelibs
```

Then run Sherpa again:

```
Sherpa
```

You might want to pipe the output to less in order to scroll through it all. To do this, use the command

```
Sherpa | less
```

The output will be a list of so-called blobs, which describe the event. An example of a signal blob is:

```
Blob [0]( 1, Signal Process      , 2 -> 2 @ (0,0,0,0)
Incoming particles :
[G] 2 u          1 (  4 ->  1) [( 5.0017e+01, 1.4917e-01, 2.8254e-01, 5.0016e+01), p^2= 1.3642e-12, m= 0.0000e+00] (500,  0)
[G] 2 ub        1 (  4 ->  1) [( 4.5008e+01, 1.0794e-01, 2.2479e-01,-4.5007e+01), p^2= 2.2737e-13, m= 0.0000e+00] (  0,500)
Outgoing particles :
[H] 2 e-        2 (  1 ->  5) [( 4.6463e+01, 3.9633e+01,-1.6547e+01,-1.7728e+01), p^2= 1.3642e-12, m= 0.0000e+00] (  0,  0)
[H] 2 e+        3 (  1 ->  5) [( 4.8562e+01,-3.9376e+01, 1.7054e+01, 2.2736e+01), p^2= 1.8190e-12, m= 0.0000e+00] (  0,  0)
```

This blob describes a  $u\bar{u} \rightarrow e^+e^-$  event. You can see that 4-momentum is conserved within a blob. This blob is linked to many other blobs, describing other interactions within the event, such as initial state radiation, parton showers, fragmentation, and hadron decays. Try to understand the structure of such an event and if you would like any more information feel free to ask one of the Sherpa authors in the room.

## 2 Underlying Event

### 2.1 Physics

The main effect of the underlying event (UE) is an increase in QCD activity. Data suggests that there are events where more than one pair of incoming partons interact during one proton-proton interaction. In other words, there is a hard component to the interactions of the remaining partons, i.e. double or multiple parton-parton interactions. This forms the basis of a very successful branch of UE models (the Pythia and Herwig ones), relying on multiple interactions. In both cases, the scale of the hardest event is used as the maximum scale for all secondary (and following) interactions, each of which is generated by textbook methods: PDFs are convolved with the matrix element squared for the secondary (QCD  $2\rightarrow 2$ ) parton-level interaction. In the case of the Pythia model the secondary interactions are subjected to the constraint that the scale associated with them is lower than the scale of the previous interaction. Thus a chain of softer and softer interactions emerges, until an infrared cut-off is reached that shields the kinematical singularities of the  $2\rightarrow 2$  scatterings.

### 2.2 Running Sherpa

Copy the prepared Run.dat into a working directory:

```
mkdir UE
cp /mt/data-grid/yeti/setups/UE/Run.dat UE/
cd UE
mkfifo fifo.hepmc2g
```

then run Sherpa and Rivet (note that this is only one command line):

```
rivetgun -a CDF_2001_S4751469 -a CDF_2004_S5839831 -H withUE -i fifo.hepmc2g -n 50000 &
  Sherpa EVENTS=50000
```

This command runs both Rivet and Sherpa simultaneously, with Rivet running in the background. For more information about why this is necessary, see the **Using Rivet with Sherpa** section of your introduction sheet.

Once this has finished, we will run the same process without the underlying event switched on. Use a text editor to modify Run.dat, for example by using the command **'nano Run.dat'**. Change the **'MLHANDLER'** from **'Amisic'** to **'None'**, to turn off the multiple interactions.

Now run Sherpa again, using the command below. Remember to change the flag for the histogram filename from **'-H withUE'** to **'-H withoutUE'**.

```
rivetgun -a CDF_2001_S4751469 -a CDF_2004_S5839831 -H withoutUE -i fifo.hepmc2g -n 50000 &
  Sherpa EVENTS=50000
```

If you have time left, you can try to play around with the impact of parameters on the underlying event activity. Starting from what you have learned in today's lecture, and reading the Sherpa manual, try to identify the parameter which will steer the amount of underlying event activity and see how this will affect your results (work in separate working directories for each parameter set!). This should give you an idea of how sensitive it is to such parameter changes (tunes), and why it is important, to systematically take as much data into account as possible, when tuning such parameters.

### 2.3 Plotting your results

To plot your results, open a terminal in your Vega window (not on the cluster node!), and enter the following two commands in your working directory (**\$DATADIR** has been predefined):

```
compare-histos $DATADIR/CDF_2001_S4751469.aida $DATADIR/CDF_2004_S5839831.aida
  withUE.aida:'w/ UE' withoutUE.aida:'w/o UE'
make-plots CDF*.dat
```

This produces plots in ".ps" files, which you can view using, for example, **'gv CDF\_2001\_S4751469.ps'** etc.